**A**

**MINI PROJECT**

**On**

**WEAPON DETECTION IN PUBLIC AREAS**

Submitted in partial fulfillment of the requirement for the award of the degree of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

By

**V.PRIYANKA (19R95A0527)**

Under the Guidance

Of

CH.RAMESH

**Asst.Professor**

**Department of CSE**



**Department of Computer Science and Engineering**

**TEEGALA KRISHNA REDDY ENGINEERING COLLEGE**

**Medbowli, Meerpet, Saroornagar, Hyderabad – 500097**

**(Affiliated to JNTUH, Approved by AICTE, Accredited by NBA)**



**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowned our efforts with success.

I extend my deep sense of gratitude to **Dr.K.Venkata Murali Mohan**, **Principal** Teegala Krishna Reddy Engineering College, Meerpet, for permitting me to undertake this technical seminar.

I am also indebted to **Dr.CH.V.Phani Krishna, Professor& Head of the Department**, Computer Science Engineering, Teegala Krishna Reddy Engineering College, Meerpet for his support and guidance throughout my technical seminar.

I am indebted to my guide. **Dr.K.Bhargavi Professor**, Computer Science Engineering, Teegala Krishna Reddy Engineering College, Meerpet for his support and guidance throughout my technical seminar.

Finally, I express thanks to one and all that have helped me in successfully completing this technical seminar. Further I would like to thank my family and friends for their moral support and encouragement.

**Submitted by**

**V.PRIYANKA (19R95A0527)**

**ABSTRACT**

**1. INTRODUCTION** …………………………………………….1

**2. LITERATURE SURVEY**………………………………...3

**3. SYSTEM ANALYSIS** ……………………………………5

3.1.Existingsystem…..........................................................................6

3.2 proposed system……………........................................................8

**4. SYSTEM DESIGN**……………………………………………..21

**5. SYSTEM IMPLEMENTATION** …………………………...25

**6. SYSTEM TESTING** …………………………………………27

**7. CODING** ……………………………………………………….31

**8. INPUT & OUTPUT SCREENS** ….………………………..35

**9. CONCLUSION**  ……………………………………………...37

**10. REFERENCES** ……………………………………………….38

.

**LIST OF FIGURES**

**Serial no System Design Page no**

**1. Use case diagram 22**

**2. Activity diagram 23**

**3. Class diagram 24**

**ABSTRACT**

Closed circuit television systems (CCTV) play a vital role in evidence collection against crimes and criminals. The existing systems does not classify normal and abnormal events leading the police tobecome more reluctant to attend the crime scenes unless there was a visual verification, either by manned patrols or by electronic images from the surveillance cameras. The Proposed work is being used for surveillance, monitoring and classifications of weapons, live tracking and many more purposes. In this work, live surveillance videos are taken for monitoring and detecting the abnormal events based on real time image processing techniques. Operations of proposed project has three processing modules, first processing module is for object detection using Convolutional Neural Networks (CNN) and second processing module will handle the classification of weapons, monitoring and alarm operations will be carried out by the third processing module.

**1.INTRODUCTION**

The crime rate across the globe has increased mainly because of the frequent use of handheld weapons during violent activity. For a country to progress, the law-and-order situation must be in control. Whether we want to attract investors for investment or to generate revenue with the tourism industry, all these needs is a peaceful and safe environment. The crime ratio because of guns is very critical in numerous parts of the world. It includes mainly those countries in which it is legal to keep a firearm. The world is a global village now and what we speak or write has an impact on the people. Even if the news they heard is crafted having no truth but as it gets viral in a few hours because of the media and especially social media, the damage will be done. People now have more depression.

and have less control over their anger, and hate speechescan get those people to lose their minds. People can be brainwashed and psychological studies show that if a person has a weapon in this situation, he may lose his senses and commit a violent activity. High incidents were recorded in past few years with the use of harmful weapons in public areas. Starting with the past year’s attacks on a couple of Mosques in New Zealand,

on March 15, 2019 at 1:40 pm, the attacker attacks the Christchurch AL-Noor Mosque during a Friday prayer killing almost 44 innocent and unarmed worshippers. On the same day just after 15 minutes at 1:55 PM, another attack happened killing seven more civilians . Active shooter incidents had also occurred in USA and then in Europe. The most significant cases were those at Columbine High School (USA, 37 victims), Andreas Broeivik's assault on Uotya Island (Norway, 179 victims) or the Charlie Hebdo newspaper attack killing 23. According to stats provided by the UNODC, among 0.1 Million people of a country, the crimes involving guns are very high i-e. 1.6 in Belgium, United States having 4.7 and Mexico with a number of 21.5. CCTV cameras play an important role to overcome this problem and are considered to be one of the most important requirements for the security aspect.CCTVs are installed in every public place today and are mainly used for providing safety, crime investigation, and other security measures for detection. CCTV footage is the most important evidence incourts. After a crime is committed, law enforcement agencies arrive at the scene and take the recording of footage with them. If we look at the surveillance system of different countries around the world, UK has about 4.5 million cameras, which are used for surveillance. Sweden has about 50000 cameras installed around 2010. The government of Poland was able to reduce drug cases by 60% andstreet fights by 40% by installing just 450 cameras in the city of Poznan. China has the world's biggest surveillance system and 170 million cameras around the nation, and these are expected to expand three times, through an additional 400 million to be connected by 2020. It took only seven minutes for Chinese officials to find and apprehend BBC reporter John Sudworth using their strong CCTV cameras network and facial recognition technology and put the criminal behind the bar. In previous years, though having surveillance cameras installed, to use them for security purposes was not an easy and dependable method. A human has to be there all the time to monitor screens. CCTV operator has to monitor 20-25 screens for 10 hours. He has to look, observe, identify, and control the situation that can be harmful to the individuals and the property. As the number of screens increases, the concentration of the person decreases considerably to monitor each screen with time. It is impossible for the person monitoring the screens to keep the same level of attention all the time. The solution to aforementioned problem is to install surveillance cameras with the ability to automatically detect weapons and raise alarm to alert the operators or security personals. However, there is not much work done on algorithms for weapon detection in surveillae cameras, and related studies are often considering concealed weapon detection (CWD), mostly using X-rays or millimeter waves images employing traditional machine learning techniques . In the past few years, deep learning in particular convolutional neural network (CNN) has given groundbreaking results in object categorizing and detection. It has achieved finest results thus far in class ical problems of image processing such as grouping, detection and localization. Instead of selecting features manually, CNN automatically learns features from given data. This article presents an automatic detection and classification method of weapons for real-time scenario using state of the art deep learning models. For real-time implementation relating the problem question of this work wallet,selfie stick in not pistol class.

“detecting weapons in real-time for potential

robbers/terrorist using deep learning”, detection and classification was done for pistol, revolver and other shot handheld weapons as in single class called pistol andrelated confusion objects such as cell phone, metal detector,

A major reason behind this was our research done on weapons used in robbery cases and it further motivated us to choose pistol and revolver as our target objects. We go through several CCTV captured robbery videos on YouTube and found that almost 95% of cases have pistol or revolver as the weapon used. With the implementation of this system, many robbery crimes, and other incidents like what happened last year in New Zealand’s Christchurch mosque could be controlled using early alarm system by alerting the operator and concerned authorities so action can be taken immediately. Gun detection in real-time is a very challenging task. As our desired object has a small size so, detecting it in an image is also very challenging in presence of other objects, especially those objects that can be confused with it. Deep learning models faced several below mentioned challenges for detection and classification task:

• The first and main problem is the data through which CNN learn its features to be used later for

classification and detection.

• No standard dataset was available for weapons.

• For real-time scenarios, making a novel dataset

* manually was a very long and time-consuming
* process.
* Labeling the desired database is not an easy task, as
* all data needs to be labeled manually.
* Different detection algorithms were used, so a labeled
* dataset for one algorithm cannot be utilized for the
* other one.
* Every algorithm requires different labeling and pre-
* processing operations for the same-labeled database.
* As for real-time implementation, detection systems
* require the exact location of the weapon so gun
* blocking or occlusion is also a problem that arises
* frequently and it could.

**2 . LITERATURE SURVEY**

Reducing the life-threatening acts and providing high security are challenging at every place. Therefore, a number of researchers have contributed to monitoring various activities and behaviors using object detection. In general, a framework of smart surveillance system is developed on three levels: firstly, to extract low-level information like features engineering and object tracking; secondly, to identify unusual human activities, behavior, or detection of any weapon; and finally, the high level is about decision making like abnormal event detection or any anomaly. The latest anomaly detection techniques can be divided into two groups, which are object-centered techniques and integrated methods. The convolutional neural network (CNN) spatial-temporal system is only applied to spatial-temporal volumes of interest (SVOI), reducing the cost of processing. In surveillance videos of complex scenes, researchers in proposed a tool for detecting and finding anomalous activities. By conducting spatial-temporal convolution layer, this architecture helps one to capture objects from both time domain and frequency domain, thereby extracting both the presence and motion data encoded in continuous frames. To do traditional functions to local noise and improve detection precision, spatial-temporal convolution layers are only implemented within spatial-temporal quantities of changing pixels. Researchers proposed anomaly-introduced learning method for detecting anomalous activities by developing multi-instance learning graph-based model with abnormal and normal bimodal data, highlighting the positive instances by training coarse filter using kernel-SVM classifier and generating improved dictionary learning known as anchor dictionary learning. Thus, abnormality is measure by selecting the sparse reconstruction cost which yields the comparison with other techniques including utilizing abnormal information and reducing time and cost for SRC.

Hu et al. have contributed in detecting various objects in traffic scenes by presenting a method which detects the objects in three steps. Initially, it detects the objects, recognizes the objects, and finally tracks the objects in motion by mainly targeting three classes of different objects including cars, cyclists, and traffic signs. Therefore, all the objects are detected using single learning-based detection framework consisting of dense feature extractor and trimodal class detection. Additionally, dense features are extracted and shared with the rest of detectors which heads to be faster in speed that further needs to be evaluated in testing phase. Therefore, intraclass variation of objects is proposed for object subcategorization with competitive performance on several datasets.

Grega et al. presented an algorithm which automatically detects knives and firearms in CCTV image and alerts the security guard or operator . Majorly, focusing on limiting false alarms and providing a real-time application where specificity of the algorithm is 94.93% and sensitivity is 81.18% for knife detection. Moreover, specificity for fire alarm system is 96.69% and sensitivity is 35.98% for different objects in the video. Mousavi et al. in carried out video classifier also referred to as the Histogram of Directed Tracklets which identifies irregular conditions in complex scenes. In comparison to traditional approaches using optical flow which only measure edge features from two subsequent frames, descriptors have been developing over long-range motion projections called tracklets. Spatiotemporal cuboid footage sequences are statistically gathered on the tracklets that move through them.

Ji et al. developed a system for security footage which automatically identifies the human behavior using convolutional neural nets (CNNs) by forming deep learning model which operates directly on the raw inputs [18]. Therefore, 3D CNN model for classification requires the regularization of outputs with high-level characteristics to increase efficiency and integrating the observations of a variety of various models.

Pang et al. presented real-time concealed various object detection under human dress in. Metallic guns on human skeleton were used for passive millimeter wave imagery which relies on YOLO algorithm on dataset of small scale. Subsequently, comparison is undertaken between Single MultiBox Detector algorithm, YOLOv3-13, SSD-VGG16, and YOLOv3-53 on PMMW dataset. Moreover, the weapon detection accuracy computed 36 frames per second of detection speed and 95% mean average precision. Warsi A et al. have contributed to automatically detecting the handgun in visual surveillance by implementing YOLO V3 algorithm with Faster Region-Based CNN (RCNN) by differentiating the number of false negatives and false positives, thus, taking real-time images and incorporating with ImageNet dataset then training it using YOLO V3 algorithm. They have compared Faster RCNN to YOLO V3 using four different videos and as a result YOLO V3 imparted faster speed in real-time environment.

Manual screening procedures for detecting concealed weapons such as handguns, knives, and explosives are common in controlled access settings like airports, entrances to sensitive buildings, and public events. The detection of weapons concealed underneath a person's clothing is an important obstacle to the improvement of the security of the general public as well as the safety of public assets like airports and buildings. It is desirable to be able to detect concealed weapons from a standoff distance, especially when it is impossible to arrange the flow of people through a controlled procedure. The goal is the eventual deployment of automatic detection and recognition of concealed weapons. It is atechnological challenge that requires innovative solutions in sensor technologies and image processing. A number of sensors based on different phenomenology as well as image processing support are being developed to observe objects underneath people's clothing. The main aim of this article is to provide a tutorial overview of these developments.

Manual screening procedures for detecting concealed weapons such as handguns,

knives, and explosives are common in controlled access settings like airports, entrances to sensi-

tive buildings, and public events. It is desirable sometimes to be able to detect concealed weapons

from a standoff distance, especially when it is impossible to arrange the flow of people through a controlled

procedure. The Concealed Weapon Detection (CWD) program was started in 1995 to address this problem

(among others) under the sponsorship of the National Institute of Justice through the administration of the

Air Force Research Laboratory in the United States. The goal is the eventual deployment of automatic d

Manual screening procedures for detecting concealed weapons such as handguns,

knives, and explosives are common in controlled access settings like airports, entrances to sensi-

tive buildings, and public events. It is desirable sometimes to be able to detect concealed weapons

from a standoff distance, especially when it is impossible to arrange the flow of people through a controlled

procedure. The Concealed Weapon Detection (CWD) program was started in 1995 to address this problem

(among others) under the sponsorship of the National Institute of Justice through the administration of the

Air Force Research Laboratory in the United States. The goal is the eventual deployment of automatic detec-

tion and recognition of concealed weapons. It is a technological challenge that requires innovative solutions

Manual screening procedures for detecting concealed weapons such as handguns,

knives, and explosives are common in controlled access settings like airports, entrances to sensi-

tive buildings, and public events. It is desirable sometimes to be able to detect concealed weapons

from a standoff distance, especially when it is impossible to arrange the flow of people through a controlled

procedure. The Concealed Weapon Detection (CWD) program was started in 1995 to address this problem

(among others) under the sponsorship of the National Institute of Justice through the administration of the

Air Force Research Laboratory in the United States. The goal is the eventual deployment of automatic detec-

tion and recognition of concealed weapons. It is a technological challenge that requires innovative solutions

Manual screening procedures for detecting concealed weapons such as handguns,

knives, and explosives are common in controlled access settings like airports, entrances to sensi-

tive buildings, and public events. It is desirable sometimes to be able to detect concealed weapons

from a standoff distance, especially when it is impossible to arrange the flow of people through a controlled

procedure. The Concealed Weapon Detection (CWD) program was started in 1995 to address this problem

(among others) under the sponsorship of the National Institute of Justice through the administration of the

Air Force Research Laboratory in the United States. The goal is the eventual deployment of automatic detec-

tion and recognition of concealed weapons. It is a technological challenge that requires innovative solutions

Manual screening procedures for detecting concealed weapons such as handguns,

knives, and explosives are common in controlled access settings like airports, entrances to sensi-

tive buildings, and public events. It is desirable sometimes to be able to detect concealed weapons

from a standoff distance, especially when it is impossible to arrange the flow of people through a controlled

procedure. The Concealed Weapon Detection (CWD) program was started in 1995 to address this problem

(among others) under the sponsorship of the National Institute of Justice through the administration of the

Air Force Research Laboratory in the United States. The goal is the eventual deployment of automatic det

**3. SYSTEM ANALYSIS**

**3.1 Process Models**

WHAT IS IMAGE PROCESSING

The term digital image processing refers to scientific and technical pursuits. Processing of an image includes improvement in its appearance and efficient representation. Image processing is widely being used today. Interest in digital image processing stems from two principle applications areas: improvement of pictorial information for human interpretation, and processing of scenes data for autonomous machine perception.

2. HOW IT IS DONE

This involves converting the visual information into discrete form suitable for computer processing. Computer processes discrete information of an image using various mathematical tools MORPHOLOGY is one of them, which offers a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size.Manual screenings procedures screens the person when the person is near the screening machine and also sometimes it gives wrong alarm indications so we are need of a technology that almost detects the weapon by scanning which achieved by imaging for concealed weapons

3. IMAGING SENSORS

These imaging sensors developed for CWD applications depending on their portability, proximity and whether they use active or passive illuminations. The different types of imaging sensors for CWD based areas be 1. INFRARED IMAGERS: Infrared imagers utilize the temperature distribution information of the target to form an image. Normally they are used for a variety of night-vision applications, such as viewing vehicles and people. The underlying theory is that the infrared radiation emitted by the human body is absorbed by clothing and then re-emitted by it. As a result, infrared radiation can be used to show the image of a concealed weapon only when the clothing is tight, thin, and stationary. For normally loose clothing, the emitted infrared radiation will be spread over a larger clothing area, thus decreasing the ability to image a weapon.

2. P M W IMAGING SENSORS:

FIRST GENERATION: Passive millimeter wave (PMW) sensors measure the apparent temperature through the energy that is emitted or reflected by sources. The output of the sensors is a function of the emissive of the objects in the MMW spectrum as measured by the receiver. Clothing penetration for concealed weapon detection is made possible by MMW sensors due to the low emissive and high reflectivity of objects like metallic guns shows a visual image of a person wearing a heavy sweater that conceals two guns made with metal and ceramics.

**3.2 SDLC Models**

**Intelligent video surveillance automatic systems**.

An automatic weapon detection system can provide the early detection of potentially violent situations that is of paramount importance for citizens security. One way to prevent these situations is by detecting the presence of dangerous objects such as handguns and knives in surveillance videos. Deep Learning techniques based on Convolutional Neural Networks can be trained to detect this type of object.

The weapon detection task can be performed by different approaches of combining a region proposal technique with a classifier, or integrating both into one model. However, any deep

learning model requires to learn a quality image dataset and an annotation according to the classification or detection tasks.

Weapon detection Open Data provides quality image datasets built for training Deep Learning models under the development of an automatic weapon detection system. Weapons datasets for image classification and object detection tasks are described and can be downloaded below. The public datasets are organized depending on the included objects in the dataset images and the target task.

**3.3 Existing system**

CCTV cameras play an important role to overcome this problem and are considered to be one of the most important requirements for the security aspect. CCTVs are installed in every public place today and are mainly used for providing safety, crime investigation, and other security measures for detection. CCTV footage is the most important evidence in courts. After a crime is committed, law enforcement agencies arrive at the scene and take the recording of footage with them. If we look at the surveillance system of different countries around the world, UK has about 4.5 million cameras, which are used for surveillance. Sweden has about 50000 cameras installed around 2010. The government of Poland was able to reduce drug cases by 60% and street fights by 40% by installing just 450 cameras in the city of Poznan. China has the world’s biggest surveillance system and 170 million cameras around the nation, and these are expected to expand three times, through an additional 400 million to be connected by 2020. It took only seven minutes for Chinese officials to find and apprehend BBC reporter John Sudworth using their strong CCTV cameras network and facial recognition technology and put the criminal behind the bar.

In previous years, though having surveillance cameras installed, to use them for security purposes was not an easy and dependable method. A human has to be there all the time to monitor screens. CCTV operator has to monitor 20–25 screens for 10 hours. He has to look, observe, identify, and control the situation that can be harmful to the individuals and the property. As the number of screens increases, the concentration of the person decreases considerably to monitor each screen with time. It is impossible for the person monitoring the screens to keep the same level of attention all the time.

The solution to aforementioned problem is to install surveillance cameras with the ability to automatically detect weapons and raise alarm to alert the operators or security personals. However, there is not much work done on algorithms for weapon detection in surveillance cameras, and related studies are often considering concealed weapon detection (CWD), mostly using X-rays or millimeter waves images employing traditional machine learning techniques. In the past few years, deep learning in particular convolutional neural network (CNN) has given groundbreaking results in object categorizing and detection. It has achieved finest results thus far in classical problems of image processing such as grouping, detection and localization. Instead of selecting features manually, CNN automatically learns features from given data.

This article presents an automatic detection and classification method of weapons for real-time scenario using state of the art deep learning models. For real-time implementation relating the problem question of this work “detecting weapons in real-time for potential robbers/terrorist using deep learning”, detection and classification was done for pistol, revolver and other shot handheld weapons as in single class called pistol and related confusion objects such as cell phone, metal detector, wallet, selfie stick in not pistol class. A major reason behind this was our research done on weapons used in robbery cases and it further motivated us to choose pistol and revolver as our target object. We go through several CCTV captured robbery videos on YouTube and found that almost 95% of cases have pistol or revolver as the weapon used. With the implementation of this system, many robbery crimes, and other incidents like what happened last year in New Zealand’s Christchurch mosque could be controlled using early alarm system by alerting the operator and concerned authorities so action can be taken immediately.

Gun detection in real-time is a very challenging task. As our desired object has a small size so, detecting it in an image is also very challenging in presence of other objects, especially those objects that can be confused with it. Deep learning models faced several below mentioned challenges for detection and classification task:

* The first and main problem is the data through which CNN learn its features to be used later for classification and detection.
* No standard dataset was available for weapons.
* For real-time scenarios, making a novel dataset manually was a very long and time-consuming process.
* Labeling the desired database is not an easy task, as all data needs to be labeled manually.
* Different detection algorithms were used, so a labeled dataset for one algorithm cannot be utilized for the other one.
* Every algorithm requires different labeling and pre-processing operations for the same-labeled database.
* As for real-time implementation, detection systems require the exact location of the weapon so gun blocking or occlusion is also a problem that arises frequently and it could occur because of self, inter-object, or background blocking.

**3.4 PROPOSED SYSTEM**

Our proposed system is further compared with the existing literature in Table [2](https://www.hindawi.com/journals/mpe/2021/9975700/tab2/). In [21], the proposed system includes CNN-based VGG-16 architecture as feature extractor, followed by state-of-the-art classifiers which are implemented on a standard gun database. The researchers investigated four machine learning models, namely, BoW, HOG + SVM, CNN, and Alexnet + SVM, to recognize the firearms and knifes from a dataset of images [22]. Their work suggests that pretrained Alexnet + SVM performed the best. As it is evident from the previous studies, researchers have widely applied CNN and its variant for weapon or knife identification from CCTV videos [23]. It is obvious from Table [2](https://www.hindawi.com/journals/mpe/2021/9975700/tab2/) that the implemented YOLO v3 outperforms the rest of the other models.

YOLOv3 (You Only Look Once, Version 3) is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images. YOLO uses features learned by a [deep convolutional neural network](https://viso.ai/deep-learning/deep-neural-network-three-popular-types/) to detect an object. Versions 1-3 of YOLO were created by Joseph Redmon and Ali Farhadi.The first version of YOLO was created in 2016, and version 3, which is discussed extensively in this article, was made two years later in 2018. YOLOv3 is an improved version of YOLO and YOLOv2. YOLO is implemented using the Keras or OpenCV deep learning libraries.

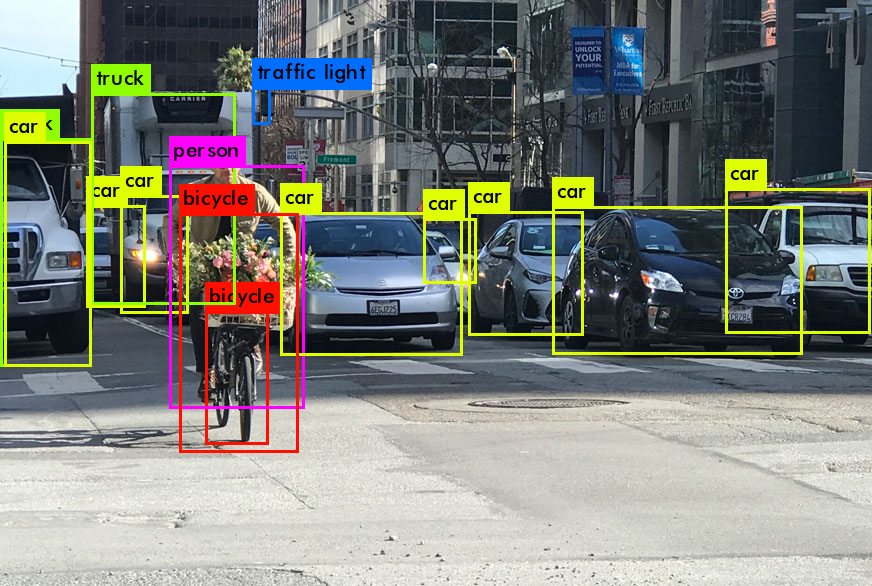


Figure :object classification in real time

Object classification systems are used by Artificial Intelligence (AI) programs to perceive specific objects in a class as subjects of interest. The systems sort objects in images into groups where objects with similar characteristics are placed together, while others are neglected unless programmed to do otherwise.

##### Why the name “you only look once”?

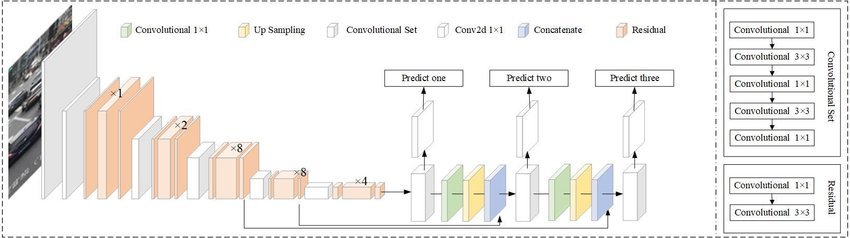
As typical for object detectors, the features learned by the convolutional layers are passed onto a classifier which makes the detection prediction. In YOLO, the prediction is based on a convolutional layer that uses 1×1 convolutions.

YOLO is named “you only look once” because its prediction uses 1×1 convolutions; the size of the prediction map is exactly the size of the feature map before it.

### How does YOLOv3 work? (Overview)

YOLO is a [Convolutional Neural Network (CNN)](https://viso.ai/deep-learning/ann-and-cnn-analyzing-differences-and-similarities/) for performing object detection in real-time. CNNs are classifier-based systems that can process input images as structured arrays of data and identify patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still maintains accuracy.

It allows the model to look at the whole image at test time, so its predictions are informed by the global context in the image. YOLO and other convolutional neural network algorithms “score” regions based on their similarities to predefined classes.

High-scoring regions are noted as positive detections of whatever class they most closely identify with. For example, in a live feed of traffic, YOLO can be used to detect different kindsof vehicles de[](https://viso.ai/wp-content/uploads/2021/02/YOLOv3-how-it-works.jpg)pending on which regions of the video score highly in comparison to predefined classes of vehicles.

**The YOLO Architecture at a Glance**

The YOLOv3 algorithm first separates an image into a grid. Each grid cell predicts some number of boundary boxes (sometimes referred to as anchor boxes) around objects that score highly with the aforementioned predefined classes.

Each boundary box has a respective confidence score of how accurate it assumes that prediction should be and detects only one object per bounding box. The boundary boxes are generated by clustering the dimensions of the ground truth boxes from the original dataset to find the most common shapes and sizes.

Other comparable algorithms that can carry out the same objective are R-CNN (Region-based Convolutional Neural Networks made in 2015) and Fast R-CNN (R-CNN improvement developed in 2017), and [Mask R-CNN](https://viso.ai/deep-learning/mask-r-cnn/).However, unlike systems like R-CNN and Fast R-CNN, YOLO is trained to do classification and bounding box regression at the same time.

What’s New in YOLOv3?

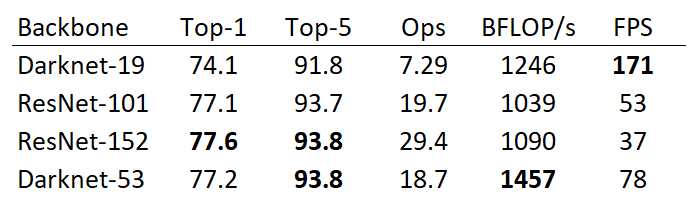
There are major differences between YOLOv3 and older versions occur in terms of speed, precision, and specificity of classes. YOLOv2 and YOLOv3 are worlds apart in terms of accuracy, speed, and architecture. YOLOv2 came out in 2016, two years before YOLO v3.

The following sections will give you an overview of what’s new in YOLOv3.

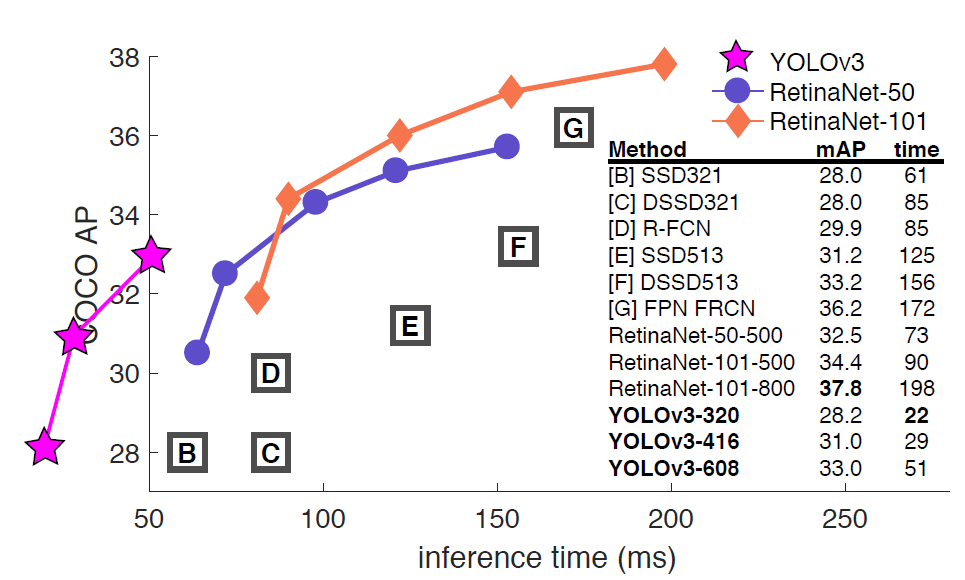
Speed

YOLOv2 was using Darknet-19 as its backbone feature extractor, while YOLOv3 now uses Darknet-53. Darknet-53 is a backbone also made by the YOLO creators Joseph Redmon and Ali Farhadi.

Darknet-53 has 53 convolutional layers instead of the previous 19, making it more powerful than Darknet-19 and more efficient than competing backbones (ResNet-101 or ResNet-152).

[](https://viso.ai/wp-content/uploads/2021/02/yolov3-comparison-performance.png)Comparison of backbones. Accuracy, billions of operations (Ops), billion floating-point operations per second (BFLOP/s), and frames per second (FPS) for various networks.

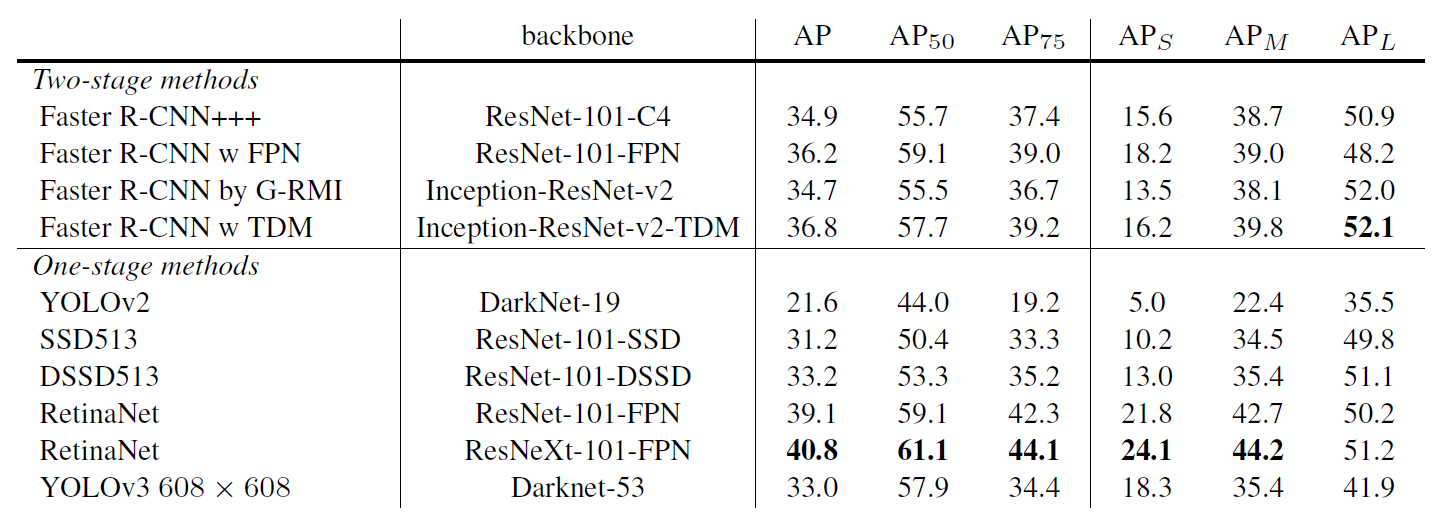
Using the chart provided in the [YOLOv3 paper by Redmon and Farhadi](https://arxiv.org/abs/1804.02767), we can see that Darknet-52 is 1.5 times faster than ResNet101. The depicted accuracy doesn’t entail any trade-off between accuracy and speed between Darknet backbones either since it is still as accurate as ResNet-152 yet two times faster.YOLOv3 is fast and accurate in terms of mean average precision (mAP) and intersection over union (IOU) values as well. It runs significantly faster than other detection methods with comparable performance (hence the name – You only look once).Moreover, you can easily trade-off between speed and accuracy simply by changing the model’s size, and no retraining required.

[](https://viso.ai/wp-content/uploads/2021/02/yolo-detection-model-comparison.png)YOLOv3 runs much faster than other detection methods with a comparable performance using an M40/Titan X GPU.

##### Precision for Small Objects

The chart below (taken and modified from the [YOLOv3 paper](https://arxiv.org/abs/1804.02767)) shows the average precision (AP) of detecting small, medium, and large images with various algorithms and backbones. The higher the AP, the more accurate it is for that variable.

The precision for small objects in YOLOv2 was incomparable to other algorithms because of how inaccurate YOLO was at detecting small objects. With an AP of 5.0, it paled compared to other algorithms like RetinaNet (21.8) or SSD513 (10.2), which had the second-lowest AP for small objects.

[](https://viso.ai/wp-content/uploads/2021/02/yolov3-and-yolov2-comparison.png)

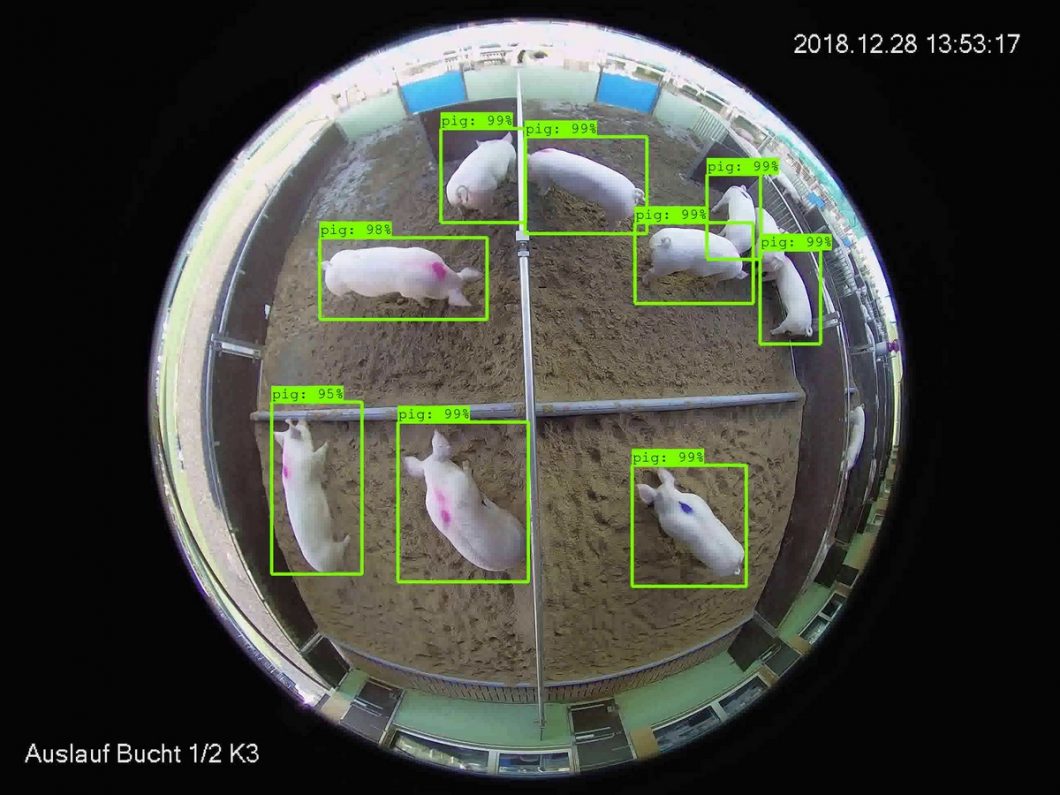
YOLOv3 comparison for different object sizes showing the average precision (AP) for AP-S (small object size), AP-M (medium object size), AP-L (large object size). YOLOv3 increased the AP for small objects by 13.3, which is a massive advance from YOLOv2. However, the average precision (AP) for all objects (small, medium, large) is still less than RetinaNet.

##### Specificity of Classes

The new YOLOv3 uses independent logistic classifiers and binary cross-entropy loss for the class predictions during training. These edits make it possible to use complex datasets such as Microsoft’s [Open Images Dataset (OID)](https://viso.ai/computer-vision/coco-dataset/) for YOLOv3 model training. OID contains dozens of overlapping labels, such as “man” and “person” for images in the dataset.

YOLO v3 uses a multilabel approach which allows classes to be more specific and be multiple for individual bounding boxes. Meanwhile, YOLOv2 used a softmax, which is a mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are proportional to the relative scale of each value in the vector.

Using a softmax makes it so that each bounding box can only belong to one class, which is sometimes not the case, especially with datasets like OID.

Object Detection to recognize animals with YOLO in a farming application

### Disadvantages of YOLOv3 vs. Other Algorithms

The YOLOv3 AP does indicate a trade-off between speed and accuracy for using YOLO when compared to RetinaNet since RetinaNet [training time is greater](https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359) than YOLOv3. However, the accuracy of detecting objects with YOLOv3 can be made equal to the accuracy when using RetinaNet by having a larger dataset, making it an ideal option for models that can be trained with large datasets.

An example of this would be common detection models like traffic detection, where plenty of data can be used to train the [model](https://viso.ai/deep-learning/ml-ai-models/) since the number of images of different vehicles is plentiful. On the other hand, YOLOv3 may not be ideal for using niche models where large datasets can be hard to obtain.

### Installing YOLOv3

The YOLOv3 installation is relatively straightforward. Installing some dependencies and libraries is necessary, and after that, it can easily be used for training models. YOLOv3 can be installed either directly onto a computer or through a notebook (such as Google Colaboratory or Jupyter). For both implementations, the commands remain the same. Assuming all libraries have been installed, the command for installing YOLOv3 is **pip install YOLOv3**.

I will briefly guide you through installing YOLOv3 with the required libraries.

1. Before installing anything, I’d advise that you make sure the pip version is at least 3.0. You can check the version with the command **pip -V**.  
   If for any reason, you are unable to uninstall older versions of pip or can’t directly use pip version 3, you can use the command “pip3 install \_\_\_” rather than just “pip.”
2. Next, we install the required libraries one by one. Starting with OpenCV (Version 3.4 or more recent):**pip install opencv-python.**
3. Python (Version 3.6 or more recent): Check if you already have python: **python –version**
4. Install Python for the first time on **Mac or Linux**: brew install python (will need homebrew first if you don’t already have it: **/bin/bash -c “$(curl -** Install Python for the first time on **Windows**: use [this guide](https://phoenixnap.com/kb/how-to-install-python-3-windows). You will need admin privileges on your computer.
5. Tensorflow-gpu (Version 1.5.0 or later): **pip install tensorflow.**
6. Keras 2.1.3: **pip install keras**Once you’ve downloaded all the above libraries, you can install YOLOv3 with the command **pip install YOLOv3.**

### How to Use YOLOv3

The first step to using YOLOv3 would be to decide on a specific object detection project. YOLOv3 performs real-time detections, so choosing a simple project that has an easy premise, such as detecting a certain kind of animal or car in a video, is ideal for beginners to get started with YOLOv3.

In this section, we will go over the essential steps and what you have to know for using YOLOv3 successfully.

##### Model Weights

Weights and cfg (or configuration) files can be downloaded from the website of the original creator of YOLOv3: <https://pjreddie.com/darknet/yolo>. You can also (more easily) use YOLO’s COCO pretrained weights by initializing the model with **model = YOLOv3().**

Using COCO’s pre-trained weights means that you can only use YOLO for object detection with any of the 80 pretrained classes that come with the [COCO dataset](https://viso.ai/computer-vision/coco-dataset/). This is a good option for beginners because it requires the least amount of new code and customization.

Object Detection with YOLO using COCO pretrained classes “dog”, “bicycle”, “truck”.

##### Making a Prediction

The convolutional layers included in the YOLOv3 architecture produce a detection prediction after passing the features learned onto a classifier or regressor. These features include the class label, coordinates of the bounding boxes, sizes of the bounding boxes, and more.

Since the prediction with YOLO uses 1 x 1 convolutions (hence the name, “you only look once”), the size of the prediction map is [exactly the size of the feature map before it](https://www.kdnuggets.com/2018/05/implement-yolo-v3-object-detector-pytorch-part-1.html).

In YOLOv3 and its other versions, the way this prediction map is interpreted is that each cell predicts a fixed number of bounding boxes. Then, whichever cell contains the center of the ground truth box of an object of interest is designated as the cell that will be finally responsible for predicting the object. There is a ton of mathematics behind the inner workings of the prediction architecture.

1. **AnchorBoxes**  
   Although anchor boxes, or bounding boxes, were discussed a little bit at the beginning of this article, there is a bit more detail about implementing them and using them with YOLOv3. Object detectors using YOLOv3 usually predict log-space transforms, which are offsets to predefined “default” bounding boxes. Those specific bounding boxes are called anchors. The transforms are later applied to the anchor boxes to receive a prediction.YOLOv3 in particular has three anchors. This results in the prediction of three bounding boxes per cell (the cell is also called a neuron in more technical terms).
2. **Non-MaximumSuppression**Objects can sometimes be detected multiple times when more than one bounding box detects the object as a positive class detection. Non-maximum suppression helps avoid this situation and only passes detections if they haven’t already been detected. Using the NMS threshold value and confidence threshold value, NMS is implemented to prevent double detections. It is an imperative part of using YOLOv3 effectively. Here, we briefly described a few of the features that make the predictions possible, such as anchor boxes and non-maximum suppression (NMS) values. This is, however, not a complete representation of all the features that go into creating a successful prediction with YOLOv3. For full descriptions of YOLOv3’s mathematical background, I suggest reading the official YOLOv3 paper linked at the end of this article.

##### Interpreting Results

Interpreting the results of a YOLO model prediction is just as nuanced as the actual implementation of the model. Multiple factors go into a successful interpretation and accuracy rating, such as the box confidence score and class confidence score used when creating a YOLOv3 computer vision model.

There are many other ways and features used when interpreting results, but these are just a few. Other YOLOv3 prediction features include the classification loss, loss function, objectness score, and more.

##### Class Confidence and Box Confidence Scores

Each bounding box has an x, y, w, h, and box confidence score value. The confidence score is the value of how probable a class is contained by that box, as well as how accurate that bounding box is.

The bounding box width and height (w and h) is first set to the width and height of the image given. Then, x and y are offsets of the cell in question and all 4 bounding box values are between 0 and 1. Then, each cell has 20 conditional class probabilities implemented by the YOLOv3 algorithm.

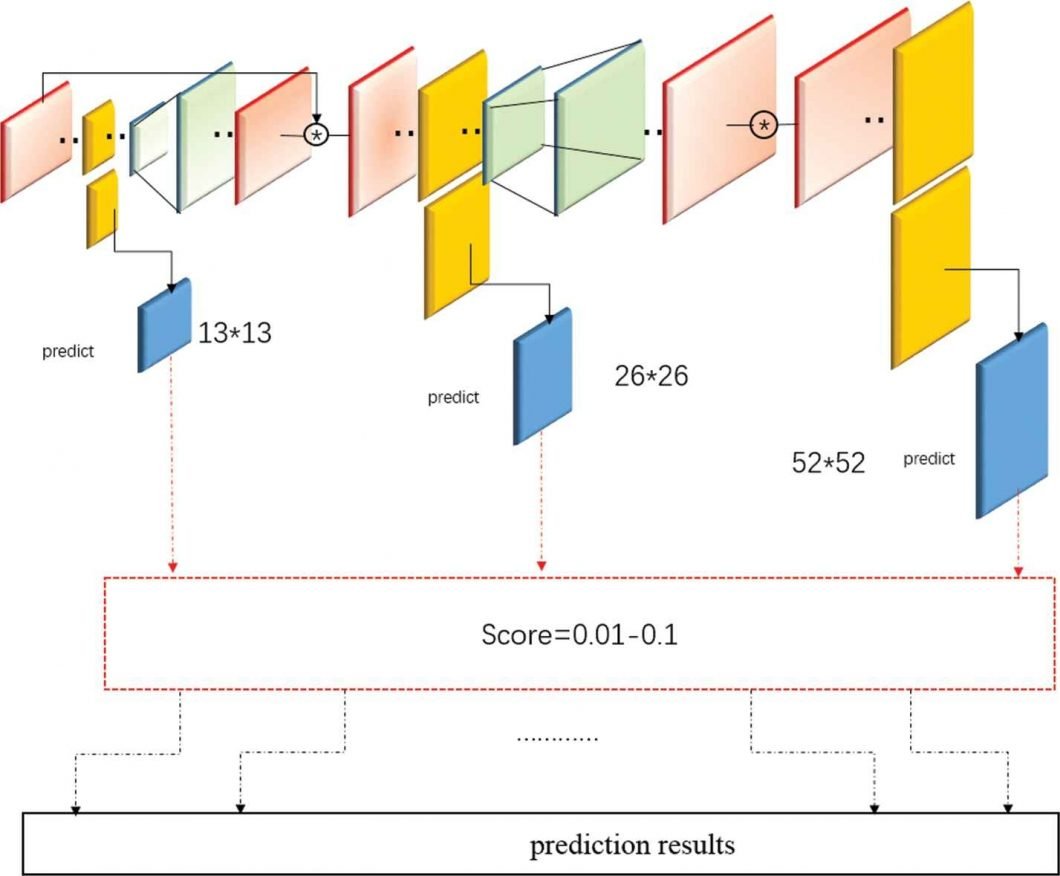
The class confidence score for each final boundary box used as a positive prediction is equal to the box confidence score multiplied by the conditional class probability. The conditional class probability in this context is the probability that the detected object is part of a certain class (the class being the object of interest’s identification). YOLOv3’s prediction, therefore, has 3 values of h, w, and depth.

There is some math that then takes place involving the spatial dimensions of the images and the tensors used in order to produce boundary box predictions, but that is complicated. If you are interested in learning what happens during this stage, I suggest the YOLOv3 Arxiv paper linked at the end of this article.

For the final step, the boundary boxes with high confidence scores (more than 0.25) are kept as final predictions . A desirable embedding can also be achieved by Adversarial Training, using the *fast gradient method* suggested in Goodfellow et al. ([2014](https://www.arxiv-vanity.com/papers/1412.6572/)). In this method, given an input x with target y, and a neural network with parameters θ, adversarial examples are generated using:

|  |  |
| --- | --- |
|  | x′=x+ϵ sign(▽xLclass(θ,x,y)) |
|  |  |

In each step an adversarial example is generated for each point x in the batch and the current parameters of the network, and classification loss is minimized for both the regular and adversarial examples. Although originally designed to improve robustness, this method seems to improve the network’s embedding for the purpose of density estimation, possibly because along the way it increases the distance between pairs of adjacent points with different labels.



### YOLOv3 Resources

The YOLOv3 algorithm has a multitude of credible resources created by the author and makers of the algorithm itself. For any purpose, primary resources are always best for getting accurate information on the topic, but for YOLO v3, these resources are even more important because of all the second-hand information available on its use – especially about newer YOLO versions (YOLOv4 and YOLOv5, that weren’t created by the original author.

In researching for this article, the most useful primary resources were:

* **YOLOv1, accredited paper on the first version of the architecture**: Redmon, Joseph, Divvala, Girshick. “You Only Look Once: Unified, Real-Time Obect Detection.”
* **YOLOv3, accredited paper on the third version of YOLO:**‌Redmon, Joseph, and Ali Farhadi. “YOLOv3: An Incremental Improvement.”
* **YOLOv3 source code and algorithm specifics by the original author (Joseph Redmon)**.
* ‌**Results from the Paper for YOLOv3:**Paperswithcode, YOLOv3: An Incremental Improvement (uploaded by Redmon and Farhadi).

### What’s Next?

YOLO is just one of many algorithms used extensively in artificial intelligence. There is another article we have written on the new version of YOLO, [YOLOv5](https://viso.ai/deep-learning/yolov5-controversy/), discussing the controversy around the new architecture and its validity.  
We suggest you check it out for more information about YOLO, as well as why the original author of YOLO did not make the new versions 4 and 5: [YOLOv5 Is Here! Is It Real or a Fake?](https://viso.ai/deep-learning/yolov5-controversy/)

[YOLOR (You Only Learn One Representation)](https://viso.ai/deep-learning/yolor/) is a different, recently released state-of-the-art object detection algorithm. The article shows how YOLOR is different from YOLO v1-v5 and why it achieves excellent performance on the [COCO](https://viso.ai/computer-vision/coco-dataset/) benchmark.

If you enjoyed reading this article, we recommend:

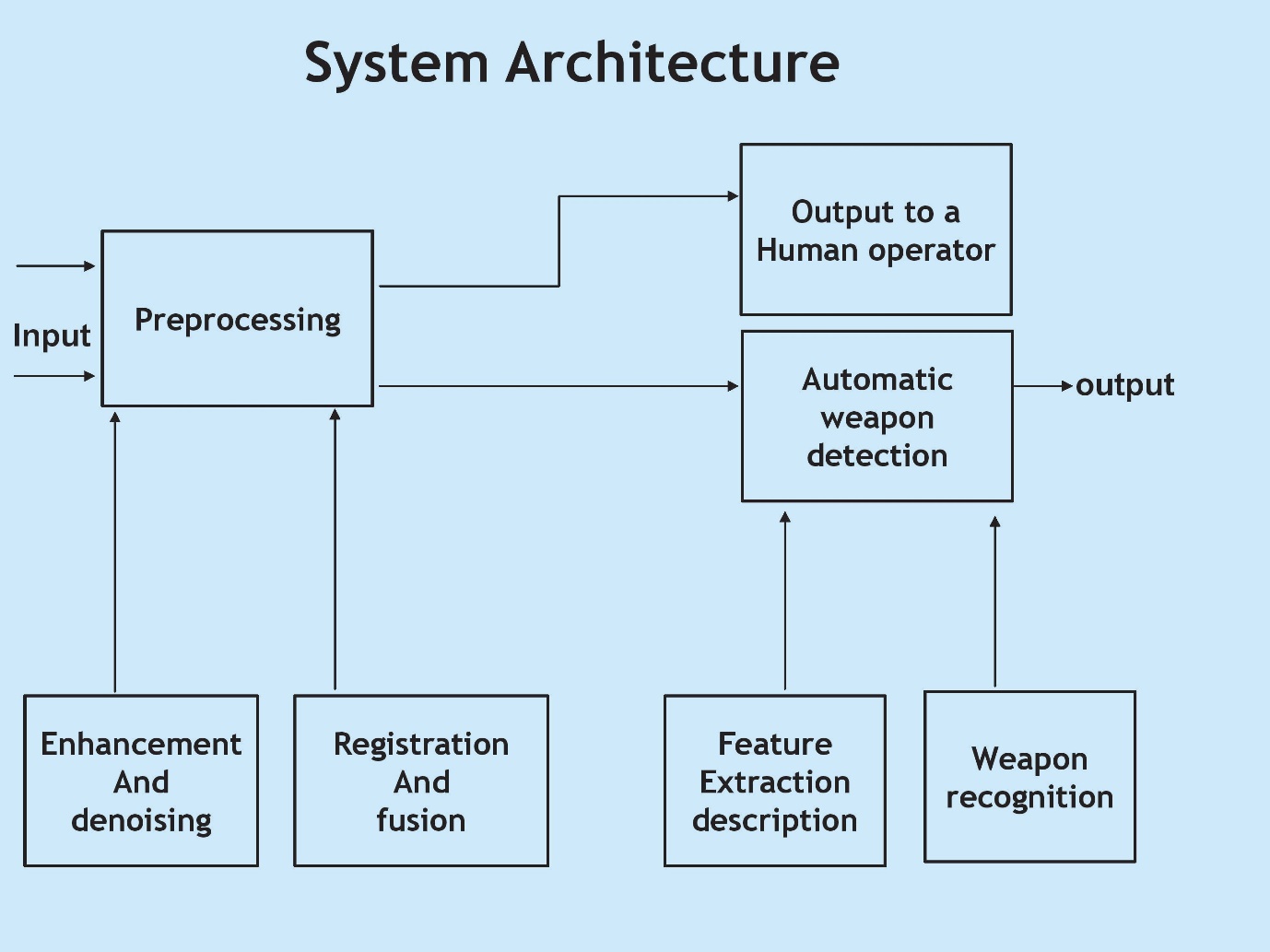
* Learn more about [Object DeteSction](https://viso.ai/deep-learning/object-detection/) and [Object Tracking](https://viso.ai/deep-learning/object-tracking/)
* Read about how using [YOLO for people counting](https://viso.ai/product/people-counting-system/) on AI hardware
* View the top [Deep Learning Frameworks](https://viso.ai/deep-learning/deep-learning-frameworks/) you need to know
* Everything you need to know about [Mask R-CNN](https://viso.ai/deep-learning/mask-r-cnn/)

**3.5 SOFTWARE AND HARDWARE REQIREMENTS**

**Hardware**: Processor: i3 ,i5 RAM: 4GB Hard disk: 16 GB • Software: operating System : Windws2000/XP/7/8/10 Anaconda,jupyter,spyder,flask Frontend :-python Backend:- MYSQL

**SOFTWARE**: Python, machine learning, yolo v3 smart survillience.

**4.SYSTEM DESIGN**

**4.1 UML DIAGRAMS**

First, an image pyramid is constructed for each source image by applying the wavelet transform to the source images. This transform domain representation emphasizes important details of the source images at different scales, which is useful for choosing the best fusion rules. Then, using a feature Selection rule, a fused pyramid is formed for the composite image from the pyramid coefficients of the source images. The simplest feature selection rule is choosing the maximum of the two corresponding transform values. This allows the integration of details into one image from two or more images. Finally, the composite image is obtained by taking an inverse pyramid transform of the composite wavelet representation. The process can be applied to fusion of multiple source imagery.

**4.2.1 USECASE DIAGRAM**

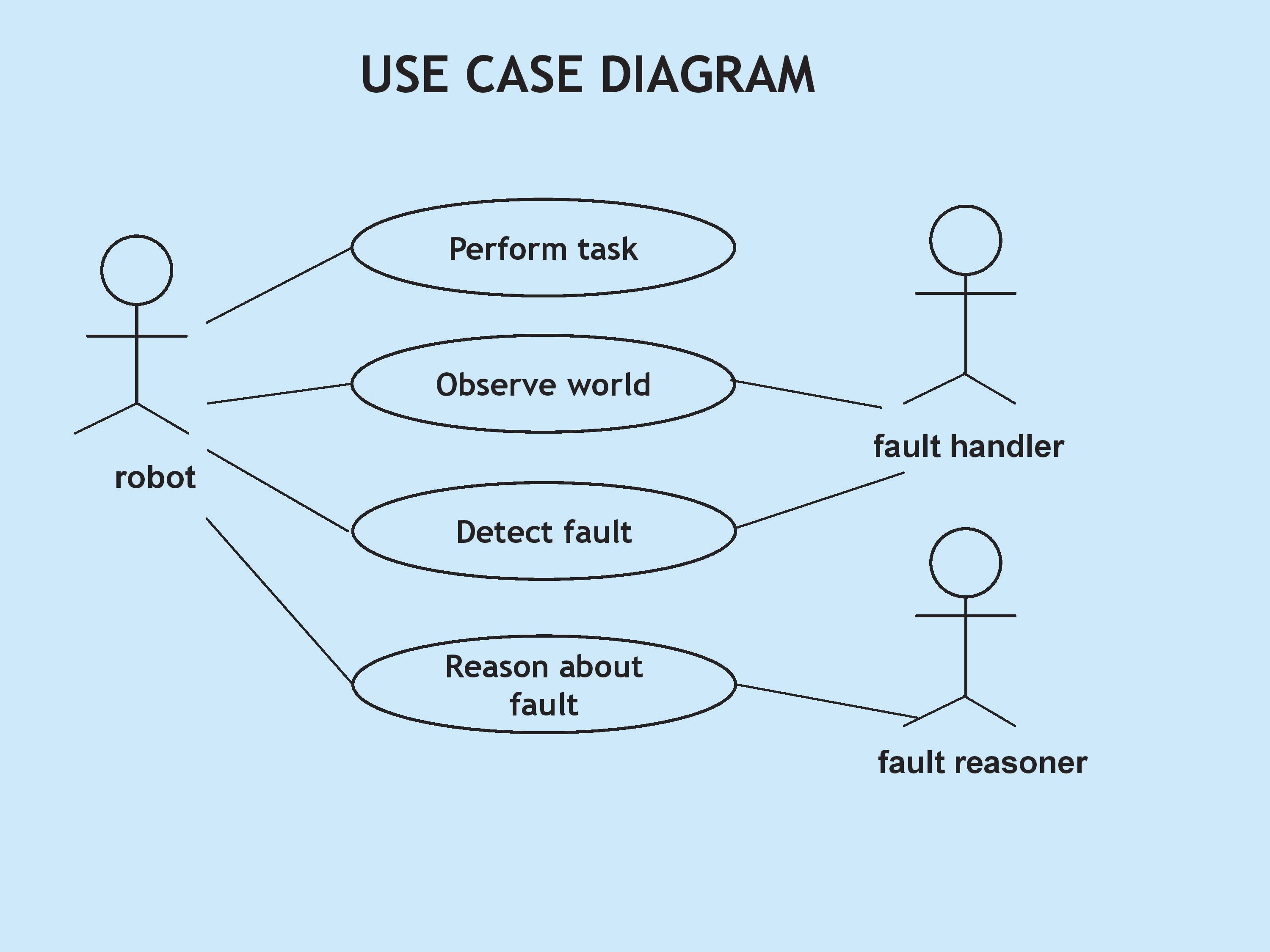


figure 1: use case diagram

**4.2.2 ACTIVITY DIAGRAM**

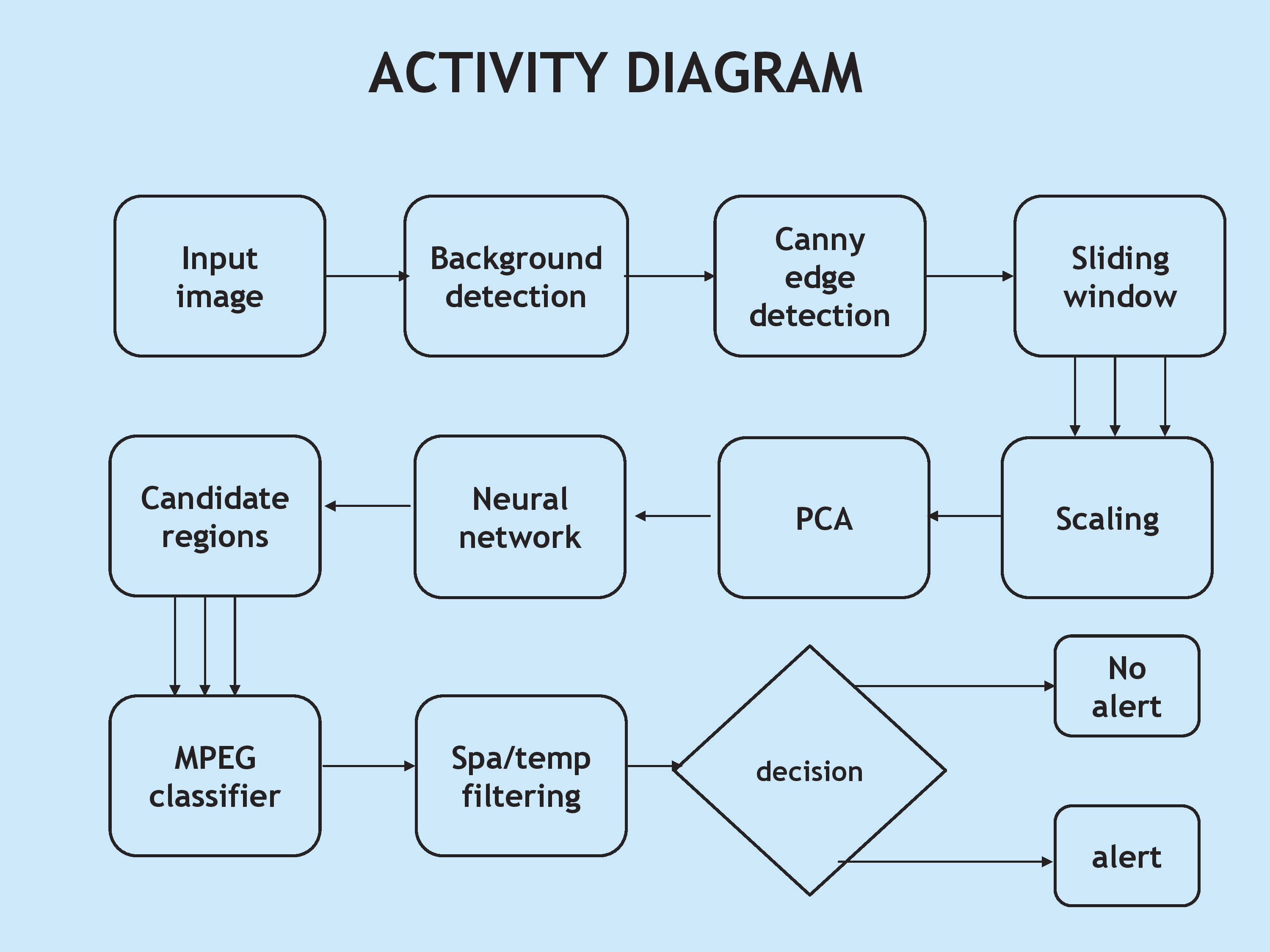


figure 2: activitiy diagram

**4.2.3 CLASS DIAGRAM**

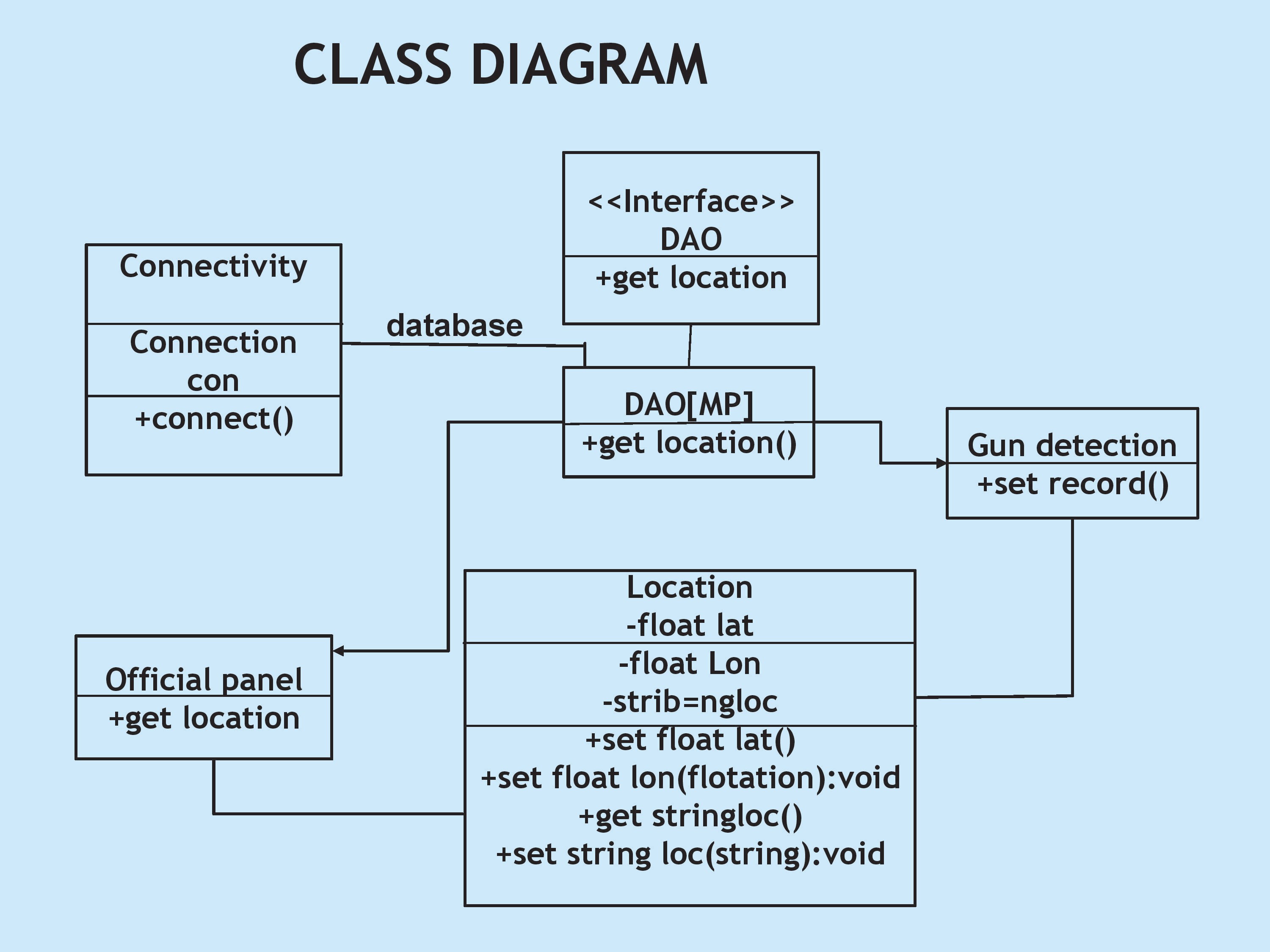


Figure 3:class diagram

**5. SYSTEM IMPLEMENTATION**

Deep learning is a branch of machine learning inspired by the functionality and structure of the human brain also called an artificial neural network. The methodology adopted in this work features the state of art deep learning, especially the convolutional neural networks due to their exceptional performance in this field. The aforementioned techniques are used for both the classification as well as localizing the specific object in a frame so both the object classification and detection algorithms were used and because our object is small with other object in background so after experimentation we found the best algorithm for our case. Sliding window/classification and region proposal/object detection algorithms were used, and these techniques will be discussed later in this section. We had started by doing the classification using different deep learning models and achieved good precision but for the real-time scenarios, the low frame per seconds of classification models were the real issue in implementation. Oxford VGG, Google Inceptionvn and InceptionResnetv2 were trained using the aforementioned approach. To achieve high precision, increase number of frame per seconds and improve localization, we moved to the object detection and region proposal methods. The different state of the art deep learning models for object detection were used and the results were compared in terms of precision, speed, and standard metric of F1 score. State of the art deep learning based SSDMobileNetv1, YOLOv3, FasterRCNN-InceptionResnetv2, and YOLOv4 were trained and tested. Different datasets were made keeping in mind the classification and detection problem as both have a separate requirement classification and localization make detection possible for any kind of detection problem giving class name as well as the region where our desired object is in the frame. A. OBJECT RECOGNITION As the name suggests, it is the process of predicting the real class or category of an image to which it belongs by making probability high only for that particular class. CNN’s are used to efficiently perform this process. Many state of the art Classification and Detection algorithms uses CNN as a backend to perform their tasks. Object Recognition to detection Hierarchy Fig. 1 depicts that classification and localization come under the category of recognition and combined classification and localization is performed to do object detection. Let us have a brief overview of the object classification, localization, and detection.

1) IMAGE CLASSIFICATION The classification model takes an image and slide the kernel/filter over the whole image to get the feature maps. From the feature extracted, it then predicts the label based on the probability.

2) OBJECT LOCALIZATION This method outputs the actual location of an object in an image by giving the associated height and width along with its coordinates

3) OBJECT DETECTION This task uses the properties of the aforementioned algorithms. The detection algorithm tells us the bounding box having x and y coordinates.

The Implementation of the weapon detection and alerting system is as follows: We utilized hand libelled data to train the model. Here to train the data in yolo we label the images first using a software called Labelimg. Using this we have manually labelled all the images from the data set. The detection of weapon is done by our yolo model. The dark net framework is used to extract features from different layers which will help us know the features of the target object. Then the bounding box is used to specify the object location and if any weapon is detected within the frame then an alert will be sent to the user through twilio api. VI. MOTION DETECTION In order to detect motion from live feed, we calculate the Pythagorean distance between two frame. Further we use standard deviation to calculate where the motion is significant enough to trigger an alarm. If the standard deviation is greater than the threshold set value, then we start the weapon detection. We initialize global variables and functions. We also used variables for setting the motion level threshold and display font. "sdThresh" is used for motion level threshold and "font" is used for setting for for text display on video. We calculate the difference between two frames and output its Pythagorean distance. Utilizing the difference between two frames we get frame 3's columns and rows matrix. We also Apply Gaussian smoothing to even out our distance mapping. hand labelled dataset was used to train our weapon detection model. After training we obtained our yolo weights file and therefore the weight file which we obtained was used for implementation of weapon detection system. If a gun is detected in the frame, an alerting system using twillio api will invoke and it will send an alert to the subscribed user.

Transfer learning has been utilized for implementing out model using YOLO V3 for weapon detection using the weights trained by us. You Only Look Once(YOLO) version 3 is algorithm is used for object detection. The given image is split into M × M mesh. An object is predicted by a cell in this mesh. Logistic regression is utilized to envision an object scores for each bounding box by YOLO V3 to compute the cost function. If a weapon is overlaid by a bounding box before more than others, the score of the resulting object should be 1. There is nil cost obtained for overlap greater than predefined threshold 0.5.

**6.SYSTEM TESTING**

This work deals with the binary classification for a real-time scenario so two classes were made and pistol and revolver images were included in pistol class and not pistol class include confusion classes like mobile phone, metal detector, selfie stick, wallet, purse, etc. For the pistol and not pistol classes, we have made three datasets, which are explained below.

#### 1) Dataset 1

This was the initial dataset used while starting this work. In this dataset, we had 1732 images in total, with 750 images in pistol class and 950 in not pistol class. Dataset was divided by the separation criteria described in Table 1 of train and test. Images were collected from online sources and database and sliding window classification algorithms were trained and tested on it.

**TABLE 1**Data Distribution

[[Table 1- 
Data Distribution](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt.t1-3059170-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt.t1-3059170-large.gif)

#### 2) Dataset 2

This was the second dataset made for the real-time scenario. This dataset contains 5254 images and classification, as well as object detection algorithms, were trained on this dataset to meet the task. Images were extracted for real-time scenario with the desired object in hand from online, sources, imfdb database, and ImageNet website. Dataset was divided by the separation criteria of test and train explained in Table 1.

#### 3) Dataset 3

This was the third dataset constructed for the real-time scenario and object detection algorithms were performed on it. This database was made by enhancing dataset 2 by overcoming the shortcomings and problems of the previous dataset. The need for this dataset arises because though we got a reasonable accuracy from classification models but the frames per second were very few. To detect images from CCTV videos, similar kinds of training data must be included so we made our own dataset to tackle this issue.This dataset contains 8327 images divided into the pistol and not pistol class. In this case, a related confusion data concept was introduced to reduce false positives and false negatives in real-time detection. Dataset images were extracted from several online sources, from CCTV videos for the particular robbery scenario, made our own dataset with a weapon in hand for the diverse scenario, did data augmentation, and finally, it was separated for test and train case.

### C. Data Distribution

Each of the aforementioned datasets are divided into the following categories mentioned in Table 1 with split size defining the separation percentage of the total data into test and train.

### D. Data Pre-Processing and Annotation

Many things affect the performance of a Machine Learning (ML) model for a specified job. First, the representation and quality of the data are essential. If there are many irrelevant and redundant data existing or noisy and unreliable data, then it is harder to discover representation during the training stage. Data preparation and filtering steps take significant processing time in ML issues [61]. The pre-processing process involves data cleaning, standardization, processing, extraction and choice of features, etc. The final training dataset is the result of pre-processing processes applied to the collected dataset.

Pre-processing is necessary for better training of a model, so the first step is to make the same size or resolution of the dataset. The next step is to apply the mean normalization. The third step is making bounding boxes on these images, which is also called annotation, localization, or labeling. In data, labeling a bounding box is made on each image. The value x, y coordinates, and width, height of the labeled object was stored in xml, csv or txt format. Following are the four main steps of data preprocessing:

* Image scaling
* Data-augmentation
* Image labeling
* Image Filtering using OpenCV

This work deals with the binary classification for a real-time scenario so two classes were made and pistol and revolver images were included in pistol class and not pistol class include confusion classes like mobile phone, metal detector, selfie stick, wallet, purse, etc. For the pistol and not pistol classes, we have made three datasets, which are explained below.

#### 1) Dataset 1

This was the initial dataset used while starting this work. In this dataset, we had 1732 images in total, with 750 images in pistol class and 950 in not pistol class. Dataset was divided by the separation criteria described in Table 1 of train and test. Images were collected from online sources and database and sliding window classification algorithms were trained and tested on it.

**TABLE 1**Data Distribution

[[Table 1- 
Data Distribution](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt.t1-3059170-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt.t1-3059170-large.gif)

#### 2) Dataset 2

This was the second dataset made for the real-time scenario. This dataset contains 5254 images and classification, as well as object detection algorithms, were trained on this dataset to meet the task. Images were extracted for real-time scenario with the desired object in hand from online, sources, and database, and ImageNet website. Dataset was divided by the separation criteria of test and train explained in Table 1.

#### 3) Dataset 3

This was the third dataset constructed for the real-time scenario and object detection algorithms were performed on it. This database was made by enhancing dataset 2 by overcoming the shortcomings and problems of the previous dataset. The need for this dataset arises because though we got a reasonable accuracy from classification models but the frames per second were very few. To detect images from CCTV videos, similar kinds of training data must be included so we made our own dataset to tackle this issue.

This dataset contains 8327 images divided into the pistol and not pistol class. In this case, a related confusion data concept was introduced to reduce false positives and false negatives in real-time detection. Dataset images were extracted from several online sources, from CCTV videos for the particular robbery scenario, made our own dataset with a weapon in hand for the diverse scenario, did data augmentation, and finally, it was separated for test and train case.

### C. Data Distribution

Each of the aforementioned datasets are divided into the following categories mentioned in Table 1 with split size defining the separation percentage of the total data into test and train.

### D. Data Pre-Processing and Annotation

Many things affect the performance of a Machine Learning (ML) model for a specified job. First, the representation and quality of the data are essential. If there are many irrelevant and redundant data existing or noisy and unreliable data, then it is harder to discover representation during the training stage. Data preparation and filtering steps take significant processing time in ML issues [61]. The pre-processing process involves data cleaning, standardization, processing, extraction and choice of features, etc. The final training dataset is the result of pre-processing processes applied to the collected dataset.

Pre-processing is necessary for better training of a model, so the first step is to make the same size or resolution of the dataset. The next step is to apply the mean normalization. The third step is making bounding boxes on these images, which is also called annotation, localization, or labeling. In data, labeling a bounding box is made on each image. The value x, y coordinates, and width, height of the labeled object was stored in xml, csv or txt format. Following are the four main steps of data preprocessing.

In order to detect motion from live feed, we calculate the Pythagorean distance between two frame. Further we use standard deviation to calculate where the motion is significant enough to trigger an alarm. If the standard deviation is greater than the threshold set value, then we start the weapon detection. We initialize global variables and functions. We also used variables for setting the motion level threshold and display font. "sdThresh" is used for motion level threshold and "font" is used for setting for for text display on video. We calculate the difference between two frames and output its Pythagorean distance. Utilizing the difference between two frames we get frame 3's columns and rows matrix. We also Apply Gaussian smoothing to even out our distance mapping.

Detecting guns in videos not only focus on gun detection but also emphasis on minimizing the number of false positives and providing real time detection. We have estimated YOLO-V3 based classification model considering a single class to specify the presence of weapon. However, this paper is to check how our YOLOv3 neural network model behaves when used in detection of weapons. The detection results are examined frame by frame in the videos during the experiments and measured a detection as true positive if the gun and bounding box overlapping is more than 50%

**7.CODING**

import cv2

import numpy as np

# Load Yolo

net = cv2.dnn.readNet("yolov3\_training\_2000.weights", "yolov3\_testing.cfg")

classes = ["Weapon"]

# with open("coco.names", "r") as f:

# classes = [line.strip() for line in f.readlines()]

layer\_names = net.getLayerNames()

output\_layers = [layer\_names[i - 1] for i in net.getUnconnectedOutLayers()]

colors = np.random.uniform(0, 255, size=(len(classes), 3))

# Loading image

# img = cv2.imread("room\_ser.jpg")

# img = cv2.resize(img, None, fx=0.4, fy=0.4)

# Enter file name for example "ak47.mp4" or press "Enter" to start webcam

def value():

val = input("Enter file name or press enter to start webcam : \n")

if val == "":

val = 0

return val

# for video capture

cap = cv2.VideoCapture(value())

# val = cv2.VideoCapture()

while True:,

img = cap.read()

height, width, channels = img.shape

# width = 512

# height = 512

# Detecting objects

blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0),

True, crop=False)

net.setInput(blob)

outs = net.forward(output\_layers)

# Showing information on the screen

class\_ids = []

confidences = []

boxes = []

for out in outs:

for detection in out:

scores = detection[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.5:

# Object detected

center\_x = int(detection[0] \* width)

center\_y = int(detection[1] \* height)

w = int(detection[2] \* width)

h = int(detection[3] \* height)

# Rectangle coordinates

x = int(center\_x - w / 2)

y = int(center\_y - h / 2)

boxes.append([x, y, w, h])

confidences.append(float(confidence))

class\_ids.append(class\_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

print(indexes)

if indexes == 0: print("weapon detected in frame")

font = cv2.FONT\_HERSHEY\_PLAIN

for i in range(len(boxes)):

if i in indexes:

x, y, w, h = boxes[i]

label = str(classes[class\_ids[i]])

color = colors[class\_ids[i]]

cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)

cv2.putText(img, label, (x, y + 30), font, 3, color, 3)

# frame = cv2.resize(img, (width, height), interpolation=cv2.INTER\_AREA)

cv2.imshow("Image", img)

key = cv2.waitKey(1)

if key == 27:

break

cap.release()

cv2.destroyAllWindows()

**8.INPUT AND OUTPUT SCREENS**

After experimentation on the previous two datasets and not finding satisfactory results for the real-time case a new dataset was made. Images were collected from robbery videos, our own dataset images holding a weapon in different scenarios, images with a dark background and low resolution, and images extracted from applying different OpenCV filters are added to make real-time detection possible. A total of 8327 images are used in this case. Following object detection models were trained and evaluated using this dataset:

* SSD MobilNetV1
* YoloV3
* Faster RCNN-Inception ResNetV2
* YoloV4

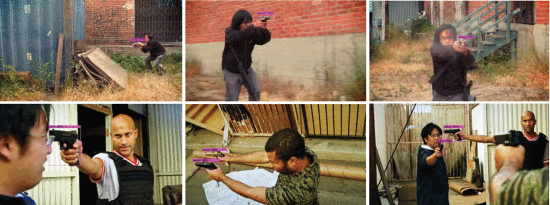
Each model had its pros and cons. SSD-MobileNet is good in terms of processing frames per second. FasterRCNN-InceptionResNetv2 has good precision and recall but not processing speed. Yolo family has a series of models. It has a different approach for the detection purpose. Unlike the other region proposal based methods, it divides the input image into an SxS grid and then simultaneously predicts the probability and bounding boxes for an object with the center falling into a grid cell. We have trained the latest state of the art Yolov3 and Yolov4 on our own weapon dataset 3 for real-time detection and best results were obtained through YOLOv4 in terms of both processing speed and precision. Table 4 below shows the results for the aforementioned detection models for this dataset at a standard threshold score of 50%.

Yolov4 performs best among all the models of both the sliding window and region proposal approach. Performance graph for yolov4 in terms of loss and mean average precision (mAP) on a validation dataset is shown in Fig. 11. We can see that how smooth is the model loss curve and how precisely it converges to the best level giving a very good loss score of 1.062 and a mean average precision of 91.73%. The mean average precision is the mean of the average precision values for all the relevant classes.

Detection Results-Top left to bottom right (a-i): (a) Image with front and side view, (b) Image vertical view (c) Image with Dark background and Low Resolution fully tilted side view, (d) Low brightness image side view slightly tilted (e) Image with the back view (f) Full front view (g) Small CCTV object (h) Very small object with side view (i) Image with full side view.

Initially, CNN architectures were quite linear. Recently, numerous variations are introduced, for example, middle blocks, skip connections, and aggregations of data between layers. These network models have already acquired rich feature representations by getting trained over a wide range of images. Thus, selecting a pretrained network and using it as a starting point to learn a new task is a concept behind transfer learning. In order to recognize the weapons, we took the weights of a pretrained model and trained another YOLO V3 model.

**Figure:**1 a pretrained model

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt16abcdef-3059170-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt16abcdef-3059170-large.gif)

**Figure:2** YOLO V3 model.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt17abcdefghi-3059170-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9353483/bhatt17abcdefghi-3059170-large.gif)

**9. CONCLUSION**

For both monitoring and control purposes, this work has presented a novel automatic weapon detection system in real-time. This work will indeed help in improving the security, law and order situation for the betterment and safety of humanity, especially for the countries who had suffered a lot with these kind of violent activities. This will bring a positive impact on the economy by attracting investors and tourists, as security and safety are their primary needs. We have focused on detecting the weapon in live CCTV streams and at the same time reduced the false negatives and positives. To achieve high precision and recall we constructed a new training database for the real-time scenario, then trained, and evaluated it on the latest state-of-the-art deep learning models using two approaches, i.e. sliding window/classification and region proposal/object detection. Different algorithms were investigated to get good precision and recall.

Through a series of experiments, we concluded that object detection algorithms with ROI (Region of Interest) perform better than algorithms without ROI. We have tested many models but among all of them, the state-of-the-art Yolov4, trained on our new database, gave very few false positive and negative values, hence achieved the most successful results. It gave 91.73% mean average precision (MAP) and a F1-score of 91% with almost 99% confidence score on all types of images and videos. We can say that it satisfactorily qualifies as an automatic real-time weapon detector. Looking at the results, we got the highest mean average precision (MAP) F1-score as compared to the research done before for real-time scenarios.The future work includes reducing the false positives and negatives even more as there is still a need for improvement. We might also try to increase the number of classes or objects in the future but the priority is to further improve precision and recall.

**10.REFERENCE**

1. Flitton, Greg, Toby P. Breckon, and Najla Megherbi. "A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery." Pattern Recognition 46, no. 9 (2013): 2420-2436.

2. Gesick, Richard, Caner Saritac, and Chih-Cheng Hung. "Automatic image analysis process for the detection of concealed weapons." In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, p. 20. ACM, 2009.

3. Tiwari, Rohit Kumar, and Gyanendra K. Verma. "A computer vision based framework for visual gun detection using harris interest point detector." Procedia Computer Science 54 (2015): 703-712.

4. Tiwari, Rohit Kumar, and Gyanendra K. Verma. "A computer vision based framework for visual gun detection using SURF." In 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), pp. 1-5. 2015

.

5. X iao, Zelong, Xuan Lu, Jiangjiang Yan, Li Wu, and Luyao Ren. "Automatic detection of concealed pistols using passive millimeter wave imaging." In 2015 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1-4. IEEE, 2015.

6. Flitton, Greg, Toby P. Breckon, and Najla Megherbi. "A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery." Pattern Recognition 46, no. 9 (2013): 2420-2436.

7. Sheen, David M.,Douglas L.McMakin, and Thomas E. Hall. "Three-dimensional millimeter-wave imaging for concealed weapon detection." IEEE Transactions on microwave theory and techniques 49, no. 9 (2001): 1581-1592.

8. Xue, Zhiyun, Rick S. Blum, and Y. Li. "Fusion of visual and IR images for concealed weapon detection." In Proceedings of the Fifth

International Conference on Information Fusion. FUSION 2002. (IEEE Cat. No. 02EX5997), vol. 2, pp. 1198-1205. IEEE, 2002.

9. Blum, Rick, Zhiyun Xue, Zheng Liu, and David S. Forsyth. "Multisensor concealed weapon detection by using a multiresolution mosaic approach." In IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004, vol. 7, pp. 4597-4601. IEEE, 2004.

10. Upadhyay, Ekta M., and N. K. Rana. "Exposure fusion for concealed weapon detection." In 2014 2nd International Conference on Devices, Circuits and Systems (ICDCS), pp. 1-6. IEEE, 2014.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI

10.1109/ACCESS.2021.3059170, IEEE

jjjjjpast year’s attacks on a couple of Mosques in New Zealand, on March 15, 2019 at 1:40 pm, the attacker attacks the Christchurch AL-Noor Mosque during a Friday prayer killing almost 44 innocent and unarmed worshippers. On

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI

10.1109/ACCESS.2021.3059170, IEEE AccessM. Tahir Bhatti et al.: Weapon Detection in Real-Time CCTV Videos using Deep Learning VOLUME XX, 2021 9 the same day just after 15 minutes at 1:55 PM, another attack happened killing seven more civilians [1]. Active shooter incidents had also occurred in USA and then in Europe. The most significant cases were those at Columbine High School (USA, 37 victims), Andreas Broeivik's assault on Uotya Island (Norway, 179 victims) or the Charlie Hebdo newspaper attack killing 23. According to stats provided by the UNODC, among 0.1 Million people of a country, the crimes involving guns are very high i-e. 1.6 in Belgium, United States having 4.7 and Mexico with a number of 21.5 [2]. CCTV cameras play an important role to overcome this problem and are considered to be one of the most important requirements for the security aspect. [3]. CCTVs are installed in every public place today and are mainly used for providing safety, crime investigation, and other security measures for detection. CCTV footage is the most important evidence in courts. After a crime is committed, law enforcement agencies arrive at the scene and take the recording of footage with them [4]. If we look at the surveillance system of different countries around the world, UK has about 4.5 million cameras, which are used for surveillance. Sweden has about 50000 cameras installed around 2010. The government of Poland was able to reduce drug cases by 60% and street fights by 40% by installing just 450 cameras in the city of Poznan [5]. China has the